

AMENDMENTS TO THE CLAIMS

This listing of claims replaces all prior versions, and listings, of claims in the application:

Listing of Claims:

15. (Currently Amended) A system that employs dynamic load balancing to asynchronously process synchronous requests, comprising:

one or more microprocessors that execute the following computer executable components stored on a ~~non-transitory~~ computer readable storage medium:

a query management component that:

receives a web-based request from a client; and

publishes the web-based request in a queue;

an asynchronous processing component that:

detects available processing engine capacity;

predicts future processing engine capacity; and

distributes portions of the web-based request among processing engines based on the detected and predicted processing engine capacity, including distributing a same portion of the web-based request to a plurality of different processing engines, such that each of the different processing engines in the plurality of processing engines returns a result for the same portion of the web-based request, whereafter a first result returned from the plurality of processing engine is initially selected for use;

an error handling component that automatically determines if the first result returned is in error and uses a subsequent result returned from the plurality of processing engines if available, and discards the first result if the first result is in error or discards subsequent results when the first result returned is not in error, or if subsequent results are not available, conveys one or more portions of the web-based request associated with a failed processing engine to another processing engine, wherein the client is not informed of a processing failure;

a process engine component that groups processing engine results;

an output component that returns the grouped processing engine results synchronous with the web-based request;

an orchestrator component that tracks and maintains one or more associations between the portions of the web-based request as the portions of the web-based request traverse through the processing engines.

16. (Previously Presented) The system of claim 15, further comprising an adapter that translates the web-based request received *via* TCP/IP, IPX/SPX UDP/IP, HTTP, SOAP, or a proprietary synchronous protocol and conveys the translated web-based request to the processing engine component through an application processing interface (API).

17. (Previously Presented) The system of claim 16, wherein the adapter is one of a pluggable software component or an instance of an object.

42. (Previously Presented) The system of claim 17, wherein the queue is utilized to store information related to a type of connection through which the web-based request was received in order to track the web-based request during processing.

44. (New) In a system that employs dynamic load balancing to asynchronously process synchronous requests, a method comprising:

receiving a web-based request from a client;

publishing the web-based request in a queue;

detecting available processing engine capacity;

predicting future processing engine capacity;

distributing portions of the web-based request among processing engines based on the detected and predicted processing engine capacity, including distributing a same portion of the web-based request to a plurality of different processing engines, such that each of the different processing engines in the plurality of processing engines returns a result for the same portion of

the web-based request, whereafter a first result returned from the plurality of processing engine is initially selected for use;

automatically determining if the first result returned is in error and using a subsequent result returned from the plurality of processing engines if available, and discarding the first result if the first result is in error or discarding subsequent results when the first result returned is not in error, or if subsequent results are not available, conveying one or more portions of the web-based request associated with a failed processing engine to another processing engine, wherein the client is not informed of a processing failure;

grouping processing engine results;

returning the grouped processing engine results synchronous with the web-based request;

and

tracking and maintaining one or more associations between the portions of the web-based request as the portions of the web-based request traverse through the processing engines.